# CHUYÊN ĐỀ
# TỔ CHỨC DỮ LIỆU
## CÁC KIỂU DỮ LIỆU CƠ BẢN

Giảng viên: VŨ QUỐC HOÀNG
(vqhoang@fit.hcmus.edu.vn)

# Nội dung trình bày



- Kiểu dữ liệu
- Phân loại kiểu dữ liệu
- Các kiểu dữ liệu cơ bản trong C/C++
- Hằng trực kiện trong C/C++
- Kiểu dữ liệu của biểu thức
- Ép kiểu

# Kiểu dữ liệu
# Kiểu dữ liệu (data type) là gì?



- Là sự phân loại dữ liệu
- Một kiểu dữ liệu xác định:
  - Các dữ liệu có thể có của kiểu
  - Tập các thao tác có thể dùng để thao tác trên dữ liệu của kiểu
  - Định dạng và cách lưu trữ dữ liệu của kiểu
- Một kiểu dữ liệu thường được định danh bằng một tên, gọi là tên kiểu dữ liệu

# Kiểu dữ liệu
# Ví dụ về kiểu dữ liệu



- Kiểu dữ liệu `int` trong C/C++ (32 bít):
  - Tên kiểu: **`int`**
  - Các giá trị: {−2147483648, …, 2147483647}
  - Các thao tác: {+, -, *, /, %, ~, &, |, …}
  - Định dạng: dạng bù-2, cơ số 2
  - Lưu trữ: 4 byte liên tiếp, byte thấp trước nếu là little-endian, byte cao trước nếu là big-endian

# Phân loại kiểu dữ liệu



- Theo mức độ trừu tượng
  - Kiểu dữ liệu máy
    - dãy bít (bit, byte, word, double word)
    - số nguyên dạng bù-2 (8/16/32/64 bít), số thực chấm động đơn/kép theo định dạng IEEE
  - Kiểu dữ liệu cụ thể: cài đặt cụ thể trong các ngôn ngữ lập trình
  - Kiểu dữ liệu trừu tượng: kiểu chung, có thể được cài đặt cụ thể thành các kiểu dữ liệu cụ thể
    - Dạng trừu tượng nhất gọi là cấu trúc dữ liệu trừu tượng (abstract data structure)

# Phân loại kiểu dữ liệu (cụ thể)



- Theo mức độ phức tạp:
  - Kiểu cơ bản (basic type): kiểu nền tảng không xây dựng từ các kiểu khác (mà giúp xây dựng các kiểu khác)
  - Kiểu phức hợp (composite type): kiểu xây dựng từ các kiểu khác
- Theo mức độ hỗ trợ:
  - Kiểu có sẵn (built-in type): kiểu do ngôn ngữ lập trình hỗ trợ sẵn
  - Kiểu do người dùng định nghĩa (user-defined type): kiểu do lập trình viên xây dựng
- Đa số kiểu cơ bản là kiểu có sẵn
- Thuật ngữ kiểu nguyên thủy (primitive type) là chỉ kiểu vừa cơ bản vừa có sẵn

# Các kiểu dữ liệu cơ bản trong C/C++



- Các kiểu dữ liệu cơ bản trong C/C++ rất gần gũi với các kiểu máy
- Các thao tác trên các kiểu này được chuyển gần như trực tiếp thành các chỉ thị máy
- Dữ liệu của các kiểu này thường được gọi là giá trị (value)
- Là nền tảng để xây dựng nên các kiểu khác

# Các kiểu dữ liệu cơ bản trong C/C++
## Các kiểu số nguyên



- Các kiểu số nguyên trong C/C++ phân ra có dấu, không dấu và theo kích thước (số bít lưu trữ)

| Kích thước | Kiểu | Phạm vi giá trị |
|---|---|---|
| 1 byte | (signed) char | $-2^7$ .. $2^7 - 1$ |
| | unsigned char | 0 .. $2^8 - 1$ |
| 2 byte | (signed) short | $-2^{15}$ .. $2^{15} - 1$ |
| | unsigned short | 0 .. $2^{16} - 1$ |
| 4 byte | (signed) int / (singned) long | $-2^{31}$ .. $2^{31} - 1$ |
| | unsigned int / unsigned long | 0 .. $2^{32} - 1$ |
| 8 byte | (signed) long long | $-2^{63}$ .. $2^{63} - 1$ |
| | unsigned long long | 0 .. $2^{64} - 1$ |

# Các kiểu dữ liệu cơ bản trong C/C++
# Các kiểu số thực



- Các kiểu số thực trong C/C++ là kiểu dấu chấm động (floating-point) theo định dạng của IEEE

| Kích thước | Từ khóa | Giá trị nhỏ nhất | Giá trị lớn nhất |
|---|---|---|---|
| 4 byte | float | $10^{-38}$ | $10^{38}$ |
| 8 byte | double | $10^{-308}$ | $10^{308}$ |
| 10 byte | long double | $10^{-4932}$ | $10^{4932}$ |

# Các kiểu dữ liệu cơ bản trong C/C++
# Kiểu kí tự và kiểu chuỗi



- C/C++ không có kiểu kí tự riêng mà dùng kiểu `char` (số nguyên 1 byte)
- Kí tự tương ứng với số nguyên qua bảng mã ASCII, là bảng tương ứng từ tập số [0, .., 255] đến tập kí tự chuẩn ASCII
- C/C++ cũng không có kiểu chuỗi riêng mà dùng mảng các kí tự kết thúc bằng kí tự `'\0'` để biểu diễn chuỗi

# Các kiểu dữ liệu cơ bản trong C/C++
# Kiểu luận lý



- C không có kiểu luận lý riêng mà dùng các giá trị số như là giá trị luận lý: 0 là false và khác 0 là true
- Các biểu thức quan hệ (==, !=, >, >=, <, <=) cho giá trị 1 ứng với true và giá trị 0 ứng với false
- C++ cung cấp kiểu luận lý `bool` nhưng được xử lý bên dưới như là kiểu số nguyên

# Hằng trực kiện trong C/C++



- Hằng (trực kiện) là phương tiện của ngôn ngữ để mô tả một giá trị xác định của một kiểu dữ liệu
- Hằng nguyên: 123, -123, …
  - Kiểu `int`
  - Thêm hậu tố `l` cho kiểu `long` và `u` cho kiểu không dấu
  - Số bát phân (cơ số 8): 0123, …
  - Số thập lục phân: 0x123, …
- Hằng thực: 12.3, 0.123e2
  - Kiểu `double`
  - Thêm hậu tố `f` cho kiểu `float` và `l` cho kiểu `long double`

# Hằng trực kiện trong C/C++



- Hằng kí tự: ‘a’, ‘\n’, …
  - Kiểu `char`: giá trị nguyên, là mã ASCII của kí tự tương ứng (‘0’ là 48)
  - Đặc tả trực tiếp mã ASCII: ‘\x30’ là 48 là ‘0’, …
  - Các kí tự đặc biệt: ‘\n’, ‘\t’, ‘\\’, …
  - Kí tự null: ‘\0’ (mã ASCII là 0)
- Hằng chuỗi: “”, “hello”, “\n”, …
  - Mảng kí tự kết thúc bằng kí tự null (‘\0’)
  - Thường lưu trữ trong vùng nhớ tĩnh của chương trình (chỉ đọc)

# Hằng trực kiện trong C/C++



- Hằng liệt kê:

```
enum boolean { false, true };
enum day { Mon = 2, Tue, … };
```

  - Mỗi hằng đại diện cho một giá trị nguyên:
    - `false` = 0, `true` = 1
    - `Mon` = 2, `Tue` = 3, …
- Hằng luận lý của C++: `true`, `false`
  - `true` là 1
  - `false` là 0

# Kiểu dữ liệu của biểu thức



- Kiểu dữ liệu của toán hạng (operand)
  - Hằng: kiểu được xác định từ qui tắc của hằng
    - 123 có kiểu là `int`
    - 123.0 có kiểu là `double`
  - Biến: kiểu được xác định từ khai báo
    - `int a;`
    - `double d;`
- Kiểu dữ liệu của biểu thức (expression)
  - Xác định qua kiểu của các toán hạng và các toán tử
    - `int a = 10;`
    - `10 + a` có kiểu là `int`, còn `(10 + a) * 10.0` có kiểu là `double`

# Ép kiểu/chuyển kiểu (Type casting/Type conversion)



- Là việc chuyển một giá trị (dữ liệu) từ kiểu này (kiểu nguồn) sang kiểu khác (kiểu đích)
- Phân loại
  - Theo mức độ:
    - Đơn giản: chặt ngắn, nới rộng
    - Phức tạp: đổi định dạng
  - Theo yêu cầu:
    - Ngầm định (implicit)
    - Tường minh (explicit)

# Ép kiểu/chuyển kiểu



- Xảy ra khi cần kiểu này lại nhận được một kiểu khác:
  - Các toán hạng khác kiểu

```
int a = 10 + 15.6; //10 chuyển thành double
```

  - Phép gán

```
int a = 10.6; //10.6 chuyển thành int
```

  - Đối số và giá trị trả về của hàm

```
int f(int v) {
    return v + 0.5; //v+0.5 là double, chuyển thành int
}
double r = f(1.5); //1.5 là double, chuyển thành int
```

# Chuyển kiểu ngầm định



- Do trình biên dịch tự động làm
- Thường không làm mất mát dữ liệu:
  - Phép dâng kiểu (type promotion): luôn đảm bảo không làm mất dữ liệu
  - Cũng có thể mất mát dữ liệu liên quan đến dấu hoặc số thực
- Khá an toàn cho chương trình

```
int a = 10 + 0.5;
//10 chuyển thành double được 10.0
//10.0 + 0.5 được 10.5
//10.5 chuyển thành int được 10
//a = 10
```

# Chuyển kiểu tường minh



- Do lập trình viên yêu cầu tường minh bằng phép ép kiểu: (kiểu) giá_trị
- Thường làm mất mát dữ liệu
- Lập trình viên tự chịu trách nhiệm

```
    double f = 3/2;
    //3/2 được 1, f=1.0
    double f = 3/(double)2;
    //chuyển kiểu tường minh 2 thành
double được 2.0
    //chuyển kiểu ngầm định 3 thành 3.0
    //3.0/2.0 được 1.5
    //f=1.5
```

20

# Hỏi và Đáp